

Robust video object tracking via Bayesian model averaging based feature fusion

Yi Dai and Bin Liu[†]

School of Computer Science and Technology, Nanjing University of Posts and
Telecommunications, Nanjing, 210023 China

Abstract

In this article, we are concerned with tracking an object of interest in video stream. We propose an algorithm that is robust against occlusion, the presence of confusing colors, abrupt changes in the object features and changes in scale. We develop the algorithm within a Bayesian modeling framework. The state space model is used for capturing the temporal correlation in the sequence of frame images by modeling the underlying dynamics of the tracking system. The Bayesian model averaging (BMA) strategy is proposed for fusing multi-clue information in the observations. Any number of object features are allowed to be involved in the proposed framework. Every feature represents one source of information to be fused and is associated with an observation model. The state inference is performed by employing the particle filter methods. In comparison with related approaches, the BMA based tracker is shown to have robustness, expressivity, and comprehensibility.

Index Terms

Bayesian model averaging, feature fusion, particle filter, video, object tracking, LBP, texture feature

[†] Correspondence author. Email: bins@ieee.org.

This work has been published online by the journal Optical Engineering since Aug. 5, 2016. So please cite it as follows
Yi Dai, Bin Liu, "Robust video object tracking via Bayesian model averaging-based feature fusion, Opt. Eng. 55(8), 083102 (2016), doi: 10.1117/1.OE.55.8.083102.

I. INTRODUCTION

Object tracking in video stream [1] is an important task in many computer vision applications such as surveillance [2], augmented reality [3], human-computer interfaces [4] and medical imaging [5]. A straightforward strategy is to detect the target and determine its position frame by frame [6]. This process ignores the temporal correlation in the sequence of frame images and thus is incapable of dealing with occlusions. An alternative strategy is to use the state space to model the underlying dynamics of the tracking system. Among the numerous state space based tracking methods, the particle filter (PF), also known as Sequential Monte Carlo (SMC) method, has obtained considerable success in various kinds of visual tracking problems. The PF methods can recursively approximate the posterior probability density function (pdf) with a set of weighted random sampled particles which evolves conforming to the state space model. The PF method is neither limited to linear systems nor requires the noise to be Gaussian [7–9].

Many PF-based visual trackers have been proposed in the literature. Most of them are based on a specific feature representation of the object. The commonly used features include but not limited to such as color [10, 11], edges [1, 11, 12], texture [12] and motion [13]. Each feature has its own pros and cons in applications. For example, a tracker using only color feature can be robust to noise and partial occlusions, but suffers from illumination changes, or the presence of confusing colors in the scene [10, 11]. Employing multiple features simultaneously via feature fusion methods can conceptually avoid the limitations of the single feature based methods [12, 13], while the existing fusion mechanisms are usually designed in an arbitrary manner without theoretical guarantees.

In addition, the success of such tracking methods requires an accurate object template [14], which is usually extracted from the first frame of the video. The tracking problem can be understood as a process of finding the region which matches the template as closely as possible in the remaining frames [14]. In many algorithms, the object template keeps invariant under the assumption of that the appearance of the object remains the same throughout the entire video. This assumption may be reasonable for a certain

period of time, but eventually it becomes no longer valid.

In this article, we introduce the concept of Bayesian model averaging (BMA) into the context of visual tracking. The BMA concept is used to fuse multi-clue information in the process of object tracking. Each clue of information is associated with one type of object feature, e.g., the color or the texture feature. A byproduct of employing BMA is shown to be an adaptive object template updating procedure, which ensures the freshness of the object template. Related multiple model based visual trackers were developed in e.g., [15, 16], while they are all based on the Markov Chain Monte Carlo (MCMC) methods, such as the Gibbs sampler, for model selection and state inference. In contrast with the existing MCMC based methods, the PF algorithm we use has significantly improved computational efficiency and less complexity in tuning parameters.

The remainder of this paper is as follows. Section 2 formulates the problem and introduces the related models. Section 3 describes the proposed BMA based feature fusion theory, along with a generic implementation of it based on the PF method. Section 4 shows the experimental results, and finally, Section 5 concludes the paper.

II. PROBLEM FORMULATION AND RELATED MODELS

In this paper, we focus on the problem of single object tracking. We formulate the tracking problem as a Bayesian state filtering task. The aim is to estimate the conditional probability $p(X_t|Y_{0:t})$ of the target state X_t at time t given the sequence of observations $Y_{0:t} = (Y_0, \dots, Y_t)$. This probability is termed the posterior distribution in the Bayesian paradigm. According to the Bayes equation, the posterior can be expressed recursively as follows

$$p(X_t|Y_{0:t}) \propto \int p(Y_t|X_t)p(X_t|X_{t-1})p(X_{t-1}|Y_{0:t-1})dX_{t-1}, \quad (1)$$

where the dynamic model $p(X_t|X_{t-1})$ governs the temporal evolution of the state X_t given the previous state X_{t-1} , and the observation likelihood model $p(Y_t|X_t)$ measures the likelihood of observing Y_t given

the state X_t .

Given the dynamic and observation models (detailed in subsections II-A and II-B, respectively), the task of estimating the posterior can be decomposed into a recursively processed prediction step [7]

$$p(X_t|Y_{0:t-1}) = \int p(X_t|X_{t-1})p(X_{t-1}|Y_{0:t-1})dX_{t-1} \quad (2)$$

and update step

$$p(X_t|Y_{0:t}) = \frac{p(Y_t|X_t)p(X_t|Y_{0:t-1})}{\int p(Y_t|X_t)p(X_t|Y_{0:t-1})dX_t}. \quad (3)$$

A. Dynamic model

Given the state vector $X_t = [x_t, y_t, v_{x,t}, v_{y,t}, h_{x,t}, h_{y,t}]$, where $[x_t, y_t]$ are the object centroid, $[v_{x,t}, v_{y,t}]$ are the corresponding velocity components and $[h_{x,t}, h_{y,t}]$ are width and height of the object area, the state evolution is defined as

$$X_t \sim \mathcal{N}(X_t|X_{t-1}, \Sigma), \quad (4)$$

where the state vector X_t is Gaussian distributed with mean vector X_{t-1} and covariance matrix Σ . This covariance matrix is determined empirically beforehand by the model designer.

B. Observation models

An observation model specifies the form of the likelihood function, which measures how likely a candidate region represents the object. In this paper, we focus on two types of observation models by extracting two sources of information from the video stream, termed the color feature and the texture feature, respectively.

1) *Color feature based observation model*: The color feature based observation model estimates color-based similarities by using a region-based color histogram. Following Nummiaro et al. [10], we calculate color histograms in the RGB space using $8 \times 8 \times 8$ bins. The histograms are produced with a function $b([x, y])$, which assigns the color at location $[x, y]$ to a corresponding bin. Given the object state X , which

defines a region covered by the object, the corresponding color distribution $p_X = \{p_X^u\}_{u=1,\dots,U}$ over that region is calculated as follows

$$p_X^u = \mathbf{C} \sum_{j=1}^J \mathbf{k} \left(\frac{\|X_c - [x_j, y_j]\|}{\sqrt{H_x^2 + H_y^2}} \right) \delta(b([x_j, y_j]) - u), \quad (5)$$

where $\delta(\cdot)$ denotes the delta function, U is the number of bins, J is the number of pixels in the region of interest, X_c is the object centroid corresponding to state X , $[H_x, H_y]$ are width and height of the region of interest, the normalization factor $\mathbf{C} = \frac{1}{\sum_{j=1}^J \mathbf{k} \left(\frac{\|X_c - [x_j, y_j]\|}{H} \right)}$ ensures that $\sum_{u=1}^U p_X^u = 1$, and \mathbf{k} is a weighting function defined to be

$$\mathbf{k}(r) = \begin{cases} 1 - r^2 & 0 \leq r < 1 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

which assigns smaller weights to the pixels that are further away from the centroid [10]. Denote p_T to be the color distribution of an object template (see Subsection III-B for details about the object template), the distance between p_X and p_T is measured based on the Bhattacharyya distance [10] as follows

$$d_{X,color} = \sqrt{1 - \rho(p_X, p_T)}, \quad (7)$$

where $\rho(p_X, p_T) = \sum_{u=1}^U \sqrt{p_X^u p_T^u}$. The color feature based likelihood model is

$$p_{color}(Y|X) = \frac{1}{\sqrt{2\pi}\sigma_{color}} \exp \left(-\frac{d_{X,color}^2}{2\sigma_{color}^2} \right), \quad (8)$$

where σ_{color} has been determined empirically to be 0.1, because it is shown to be able to accommodate different scenarios in our experiments.

2) *Texture feature based observation model:* Here we focus on a local binary pattern (LBP) operator for describing texture feature. The LBP operator has been widely used in various applications such as face recognition [17]. This operator has proven to be highly discriminative, computationally efficient and invariant to monotonic gray-level changes.

The LBP operator assigns a label to every pixel of an image by thresholding the 3×3 -neighborhood of each pixel with the center pixel value. The histogram of the labels is used as a texture descriptor. A basic conceptual illustration of the LBP operator is shown in Fig.1, in which the center pixel is labeled by 154.

Let $l(x, y)$ denote the label of the pixel located at $[x, y]$. A histogram of the labeled region of interest, specified by the object state X , can be defined as follows [18]

$$H_{X,i} = \sum_{x,y} I\{l(x, y) = i\}, i = 0, 1, \dots, n - 1 \quad (9)$$

where n denotes the number of labels generated by the LBP operator and

$$I\{A\} = \begin{cases} 1, & A \text{ is true} \\ 0, & A \text{ is false} \end{cases} \quad (10)$$

A basic LBP operator with $n = 256$ is adopted here.

Denote H_0 to be a LBP generated histogram corresponding to the object template (see Subsection III-B for details about the object template). The distance between H_X and H_0 is measured based on the Bhattacharyya distance [10] as follows

$$d_{X,texture} = \sqrt{1 - \rho(H_X, H_0)}, \quad (11)$$

where $\rho(H_X, H_0) = \sum_{i=1}^n \sqrt{H_{X,i} H_{0,i}}$. The texture feature based likelihood model is just

$$p_{texture}(Y|X) = \frac{1}{\sqrt{2\pi}\sigma_{texture}} \exp\left(-\frac{d_{X,texture}^2}{2\sigma_{texture}^2}\right), \quad (12)$$

where $\sigma_{texture}$ has been determined empirically to be 0.1 here, as it can accommodate different scenarios in our experiments.

III. THE PROPOSED BMA BASED FEATURE FUSION APPROACH TO VIDEO OBJECT TRACKING

The BMA strategy is a generic solution to deal with model uncertainty problems in a Bayesian statistical paradigm [19–21]. Here we propose a BMA based feature fusion theory, along with a generic implementation of it based on the PF method, in the context of visual tracking.

A. BMA based feature fusion

We focus on the situation where several candidate features (such as the color and texture features presented in Section II-B) are available for use, but there is uncertainty on the best feature to use at each time step. Associating each feature with a plausible model, the BMA strategy is used to balance usages of the candidate models in a theoretically sound manner.

Let $\mathcal{H}_t = m$ denote the event that the m th model, \mathcal{M}_m , is the best for use at time step t . Based on BMA [19–21], the posterior distribution, as shown in Eqns. (1) and (3), can be calculated as follows

$$\begin{aligned} p(X_t|Y_{0:t}) &= \sum_{m=1}^M p(X_t|\mathcal{H}_t = m, Y_{0:t})p(\mathcal{H}_t = m|Y_{0:t}) \\ &= \sum_{m=1}^M p_m(X_t|Y_{0:t})\pi_{t|t,m} \end{aligned} \quad (13)$$

where $p_m(X_t|Y_{0:t}) \triangleq p(X_t|\mathcal{H}_t = m, Y_{0:t})$, $\pi_{t|t,m} \triangleq p(\mathcal{H}_t = m|Y_{0:t})$ and M is the number of candidate models. Here we only consider two candidate models, namely the color and texture feature based observation models as presented in subsection II-B; so we have $M = 2$. Note that all the calculations presented in what follows are valid for any value of M , $M \in \mathbb{R}^+$.

We use the PF method [7, 8] to calculate Eqn.(13). Assume that at time $t-1$, we have at hand $\pi_{t-1|t-1,m}$ and a weighted sample set, $\{X_{t-1}^i, \omega_{m,t-1}^i\}, i = 1, 2, \dots, N$, which can build up a discrete probability distribution that approximates $p_m(X_{t-1}|Y_{0:t-1})$ as follows

$$p_m(X_{t-1}|Y_{0:t-1}) \simeq \sum_{i=1}^N \omega_{m,t-1}^i \delta(X_{t-1} - X_{t-1}^i), \quad (14)$$

where $\omega_{m,t-1}^i > 0, i = 1, 2, \dots, N$ and $\sum_{i=1}^N \omega_{m,t-1}^i = 1$, for $\forall m$. Then the posterior at time $t - 1$ can be approximated as follows

$$\begin{aligned} p(X_{t-1}|Y_{0:t-1}) &= \sum_{m=1}^M p_m(X_{t-1}|Y_{0:t-1}) \pi_{t-1|t-1,m} \\ &\simeq \sum_{m=1}^M \pi_{t-1|t-1,m} \sum_{i=1}^N \omega_{m,t-1}^i \delta(X_{t-1} - X_{t-1}^i). \end{aligned} \quad (15)$$

Comparing Eqn.(15) with Eqn.(13), we can observe that, upon the arrival of Y_t , the task of calculating $p(X_t|Y_{0:t})$ can be decomposed into the following two sub-tasks,

- sub-task I: given $\{X_{t-1}^i, \omega_{m,t-1}^i\}, i = 1, 2, \dots, N$, how to generate another weighted sample set $\{X_t^i, \omega_{m,t}^i\}, i = 1, 2, \dots, N$, which can provide a Monte Carlo approximation to $p_m(X_t|Y_{0:t})$ as follows,

$$p_m(X_t|Y_{0:t}) \simeq \sum_{i=1}^N \omega_{m,t}^i \delta(X_t - X_t^i), m = 1, \dots, M. \quad (16)$$

- sub-task II: given $\pi_{t-1|t-1,m}$, how to derive $\pi_{t|t,m}$ out, for $\forall m$.

In what follows, we present solutions to these sub-tasks.

1) *PF based solution to sub-task I:* For any, say the m th, candidate model, \mathcal{M}_m , here the concern is, given $\{X_{t-1}^i, \omega_{m,t-1}^i\}, i = 1, 2, \dots, N$ that satisfies Eqn.(14), how to generate another weighted sample set $\{X_t^i, \omega_{m,t}^i\}, i = 1, 2, \dots, N$, which should satisfy Eqn.(16).

Within the PF algorithm framework, the new state samples X_t^i are first drawn from a proposal distribution $q(X_t|X_{1:t-1}, Y_{1:t})$ and then weighted according to the importance sampling strategy[7, 8]. Here the state transitional prior, as defined in Eqn.(4), is selected as the proposal, i.e., $q(X_t|X_{1:t-1}, Y_{0:t}) = p(X_t|X_{t-1})$. This type of proposal has been widely adopted in PF methods such as the condensation method and the bootstrap filter [9, 22]. Based on Eqn.(4), we generate $X_t^i, i = 1, \dots, N$ as follows

$$X_t^i \sim \mathcal{N}(X_t|X_{t-1}^i, \Sigma), i = 1, \dots, N. \quad (17)$$

The corresponding importance weights are calculated as follows[7]

$$\begin{aligned}\hat{\omega}_{m,t}^i &= \omega_{m,t-1}^i \frac{p_m(Y_t|X_t^i)p(X_t^i|X_{t-1}^i)}{q(X_t^i|X_{1:t-1}^i, Y_{0:t})} \\ &= \omega_{m,t-1}^i p_m(Y_t|X_t^i), i = 1, \dots, N.\end{aligned}\quad (18)$$

$$\omega_{m,t}^i = \frac{\hat{\omega}_{m,t}^i}{\sum_{j=1}^N \hat{\omega}_{m,t}^j}, i = 1, \dots, N. \quad (19)$$

As the operators for generating and weighting samples belong to the routine PF framework, the results have theoretical guarantees as proved in the literature [23, 24].

2) *Solution to sub-task II:* Here we focus on the following task, namely, given $\pi_{t-1|t-1,m}$, how to derive $\pi_{t|t,m}$ out, for $\forall m$. First, we consider the prediction of the model indicator, namely given \mathcal{H}_{t-1} , how to predict \mathcal{H}_t . We specify the model transition process in term of forgetting [25]. Denote α as a forgetting factor satisfying $0 < \alpha < 1$. Given $\pi_{t-1|t-1,m}$, $\pi_{t|t-1,m} \triangleq p(\mathcal{H}_t = m|Y_{0:t-1})$ is calculated as follows

$$\pi_{t|t-1,m} = \frac{\pi_{t-1|t-1,m}^\alpha}{\sum_{l=1}^M \pi_{t-1|t-1,l}^\alpha}. \quad (20)$$

Then, employing the Bayes' rule we have

$$\pi_{t|t,m} = \frac{\pi_{t|t-1,m} p_m(Y_t|Y_{0:t-1})}{\sum_{l=1}^M \pi_{t|t-1,l} p_l(Y_t|Y_{0:t-1})}, \quad (21)$$

where $p_m(Y_t|Y_{0:t-1})$ is the marginal likelihood of \mathcal{M}_m at time t , which is defined to be

$$p_m(Y_t|Y_{0:t-1}) = \int p_m(Y_t|X_t) p_m(X_t|Y_{0:t-1}) dX_t. \quad (22)$$

The element $p_m(X_t|Y_{0:t-1})$ in Eqn.(22) can be estimated as follows,

$$p_m(X_t|Y_{0:t-1}) \approx \sum_{i=1}^N \omega_{m,t-1}^i \delta(X_t - X_t^i), \quad (23)$$

where the weighted sample set $\{\omega_{m,t-1}^i, X_t^i\}, i = 1, 2, \dots, N$ is a byproduct of the PF solution to sub-task I mentioned above. Therefore we can estimated the integral in Eqn.(22) as follows

$$p_m(Y_t|Y_{0:t-1}) \simeq \sum_{i=1}^N \omega_{m,t-1}^i p_m(Y_t|X_t^i). \quad (24)$$

B. Updating object template

Both the definitions of the color and texture feature based observation models require a pre-determined object template, as shown in Eqns.(7) and (11), respectively. Based on the assumption of that the appearance of the object remains the same throughout the entire video, an invariant object template is used in many methods. This assumption may be reasonable for a certain period of time, but eventually the template will become no longer an accurate model of the appearance of the object.

Here we show that, as a byproduct of employing BMA for object tracking, an adaptive template updating mechanism can be easily realized to ensure that the current template accurately represents the new image of the object.

In our methods, the initial object template is produced by an object detector [6]. This detector employs an adaptive Gaussian mixture to model the time-evolving scene in the video stream . During the follow-up tracking process, the color or texture distribution of a predicted object region is enforced to be compared with that of the object template to determine the likelihood of the new observation via Eqn.(8) or (12). If an abrupt change in one feature space of the object happens, it will result in a sudden slump in the corresponding likelihood. Then the marginal likelihood and posterior probability of that feature based observation model, in terms of Eqn.(22) or Eqn.(21), will be reduced correspondingly. Therefore, the state estimate produced by using that feature will be assigned an extremely small probability weight in generating the final state estimate by Eqn.(13). Therefore, the BMA based method can be robust to the failure of a single feature based model in yielding accurate estimation of the object state.

The template updating procedure can be simple as follows. If a slump in the posterior probability of a feature is observed, we consider it as an indication of that we need to update the object template. We

extract a new image of the object based on the output of the BMA based tracker, and then construct a new object template model. If it is the color (or texture) feature based observation model that fails, we construct the new object template model by calculating the feature distribution by Eqn.(5)(or Eqn.(9)).

C. Implementation of the proposed algorithm

Here a particle based implementation of the proposed method is summarized as follows in **Algorithm**

1. In this implementation two models are considered and thus $m \in \{1, 2\}$, where the figures 1 and 2 correspond to the color and texture feature models, respectively.

Algorithm 1: One iteration of the proposed BMA-PF algorithm

- 1 Input: the 'old' sample set $\{\omega_{1,t-1}^i, \omega_{2,t-1}^i, X_{t-1}^i\}, i = 1, 2, \dots, N$, and the 'old' posterior probabilities of the candidate models $\pi_{t-1|t-1,1}$ and $\pi_{t-1|t-1,2}$, at time step $t - 1$; the color distribution p_T and the LBP generated histogram H_0 of the object template;
 - 2 **for** $i = 1, \dots, N$ **do**
 - 3 Sample X_t^i using Eqn.(17);
 - 4 Calculate the importance weights $\hat{\omega}_{m,t}^i, m = 1, 2$ using Eqn.(18), in which $p_1(Y_t|X_t^i)$ and $p_2(Y_t|X_t^i)$ are replaced with $p_{color}(Y_t|X_t^i)$ (defined by Eqn.(8)) and $p_{texture}(Y_t|X_t^i)$ (defined by Eqn.(12)), respectively;
 - 5 Normalize the importance weights using Eqn. (19), and get $\omega_{m,t}^i, i = 1, 2, \dots, N, m = 1, 2$;
 - 6 Calculate $\pi_{t|t-1,m}, m = 1, 2$, using Eqn. (20);
 - 7 Calculate $\pi_{t|t,m}, m = 1, 2$, using Eqns. (21-24), in which $p_1(Y_t|X_t^i)$ and $p_2(Y_t|X_t^i)$ are replaced with $p_{color}(Y_t|X_t^i)$ (defined by Eqn.(8)) and $p_{texture}(Y_t|X_t^i)$ (defined by Eqn.(12)), respectively;
 - 8 Estimate, if desired, moments of the tracked position at time step t as
 $\xi(f(X_t)) = \sum_{m=1}^M \pi_{t|t,m} \sum_{i=1}^N \omega_{m,t}^i f(X_t^i)$, obtaining, for instance, a mean position using $f(X) = X$;
 - 9 If the condition of updating the object template satisfies, update p_T or H_0 correspondingly, based on the current estimate of the object state, see details in Subsection III-B;
 - 10 Output: $\xi(f(X_t)), \{\omega_{1,t}^i, \omega_{2,t}^i, X_t^i\}, i = 1, 2, \dots, N, \pi_{t|t,1}, \pi_{t|t,2}, p_T$ and H_0 .
-

IV. EXPERIMENTAL RESULTS

We applied the proposed method to analyze real video stream data. The purpose is to demonstrate that the proposed BMA based feature fusion method really works.

Several competitor algorithms, including ALG I (PF using adaptive color feature [10]), ALG II (PF using LBP texture feature [18]), ALG III (PF using both the texture and color features, which are equally weighted [26]) and ALG IV (adaptive GM detector [6]), are involved for performance comparison. The particle size N in each algorithm is set equally to be 200.

A. Case I

The first video stream under investigation is taken by a camera placed at a fixed location in a dark tunnel. The window size of the video keeps fixed as the object appears nearer or further in the frames. The task is to track a moving car passing through the tunnel. The color of the car is similar with the road in the video. In the last frames, the taillights of the car light up leading to a change in the color feature distribution of the object.

The result of an example run of the proposed algorithm is presented in Fig.2. As is shown, the tracking result was not influenced by the presence of confusing colors, abrupt changes in the object color feature distribution and changes in scale. The change in the object template has been indicated by the change in the size of the white box, which means the object contour outputted by the algorithm. For comparison, the tracking results corresponding to the competitor algorithms are shown in Figs. 3-6, respectively. We see that, among the competitor algorithms, ALG I and II did not adapt well to the change in scale. ALG I failed to track after that the taillights of the car light up. ALG III and IV provide satisfactory tracking result.

A numerical performance comparison based on 100 times independent runs of each involved algorithm is conducted and the result is shown in Fig.7. We can see that the proposed algorithm performs best, while the performance of ALG I [10] gets deteriorated remarkably since the car's taillights light up. The computation time required to run each algorithm is presented in Table I.

The feature fusion result can be revealed by the changes in the posterior probabilities of the color and texture based observation models. Fig.8 shows the real-time output of the posterior probabilities of those two models. It is shown that the impact of the red light on the color and texture clues is significant, and that the feature fusion effect of our algorithm is truly taking effect in dynamically adjusting the usages of the color and texture features in the tracking process.

B. Case II

To further demonstrate the performance of the proposed method, we applied it to analyze another video stream. The related camera was placed at a fixed location, so the window size of the video keeps fixed as the object appears nearer or further in the frames. The object to be tracked is a green helicopter, which is controlled to fly up and down. Four typical frames along with the tracking result provided by our algorithm are listed in Fig.9. In the upper left sub-figure, we see that the helicopter is flying above the trees. Then it flies downwards and then gets partially occluded by the trees in the upper right sub-figure. The helicopter rises up again and then appears in the lower left sub-figure. Then it falls downwards again, with its body mixed up with the trees behind it in the video.

In this case, the proposed algorithm worked very well in tracking the object accurately from beginning to end. See Fig.9 for 4 typical frames in the tracking period. As shown in Fig. 10, ALG I [10] failed when the body of the object is totally mixed up with the trees in the lower right sub-figure, because the algorithm erroneously identified the crown of a tree as the object. For ALG II [18], an early tracking failure occurred when the object approaches the trees in the first time, see the upper right sub-figure of Fig. 11. In Fig.12, we see that ALG III [26] performed satisfactorily for this case.

C. Case III: PETS 2015 dataset

To further properly evaluate the proposed method, we applied it to analyze an open source dataset released in the 2015 International Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2015) [27]. We selected one dataset that conforms to the application scenario of the presented methods. This dataset is called P5, whose acronym stands for the EU project 'Privacy Preserving Perimeter Protection Project'.

The object to be tracked is a vehicle driving across the scene. This vehicle appears from the bottom right corner of the scene. It first turns right along with a riverside path. Then the vehicle moves ahead to the direction far away from the camera. The shape, contour and size of the object continue to change

over time in this video. As object's relative position changes, the object's color (especially the color of its roof) also changes gradually over time. The background is complex, containing a wide blue river, green trees, yellow grasses and some other entities like boats, rocks, and so on.

The aforementioned factors together constitute a big challenge for tracking the object accurately online, while the proposed algorithm again gives a satisfactory performance, since it tracks the object very accurately from beginning to end. See Fig.13 for the tracking results corresponding to the 300th, 340th, 420th and 460th frames of the video.

A Monte Carlo based numerical performance comparison was also conducted. Every algorithm under consideration was ran 100 times. The result is presented in Fig. 14. As is shown, the adaptive GM detector [6] performs best in the beginning 40 frames, while since the 340th frame when the route of the object begins to turn, the proposed algorithm gives the best performance in tracking accuracy in most of the time. Fig. 15 presents the posterior probabilities of the color and texture feature models, provided by a typical run of the proposed algorithm. The computation time required to run each algorithm is presented in Table II.

V. CONCLUSION

In this paper, we propose a BMA based feature fusion approach, along with a generic implementation based on the PF methods, for tracking a moving object of interest in a video stream. BMA is a theory in Bayesian statistics for dealing with model uncertainty problems. Here we use it to fuse multi-clue information of the object in dealing with complex visual tracking tasks. In theory, the BMA framework allows any number of features to be involved, while as an instantiation, an algorithm that only fuses the color and LBP based texture features is implemented here. We test the performance of the proposed algorithm with real datasets, including the P5 dataset used by PETS 2015 challenge. Our algorithm is shown to be robust against partial occlusion, presence of confusing colors, abrupt changes in the object features and changes in scale. The experimental results show that our algorithm beats several existing competitor algorithms in tracking accuracy with comparable computing burdens. In summarize,

we demonstrate that the BMA theory can provide an efficient as well as theoretically sound solution to fuse multi-clue information in visual object tracking.

In this paper, we only use the color and LBP based texture features. Many other possible combinations of differing features can be investigated and used within the BMA framework. It is also feasible to extend the reported method here to handle multi-object visual tracking.

VI. ACKNOWLEDGEMENT

This work was partly supported by the National Natural Science Foundation (NSF) of China under grant Nos. 61302158 and 61571238, the NSF of Jiangsu province under grant No. BK20130869 and the China postdoctoral Science Foundation under grant No. 2015M580455.

REFERENCES

- [1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM computing surveys (CSUR)*, vol. 38, no. 4, p. 13, 2006.
- [2] B. Benfold and I. Reid, "Stable multi-target tracking in real-time surveillance video," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2011, pp. 3457–3464.
- [3] X. Yang, X. Si, T. Xue, L. Zhang, and K. T. Cheng, "Vision-inertial hybrid tracking for robust and efficient augmented reality on smartphones," in *Proc. of the 23rd Annual ACM Conference on Multimedia Conference*. ACM, 2015, pp. 1039–1042.
- [4] D. Popa, V. Gui, and M. Ottesteanu, "Real-time multi-cue finger tracking for human computer interaction," in *Proc. of 38th International Conf. on Telecommunications and Signal Processing (TSP)*. IEEE, 2015, pp. 1–7.
- [5] M. Tinguely, O. Matar, and V. Garbin, "Tracking the deformation of a tissue phantom induced by ultrasound-driven bubble oscillations," in *Journal of Physics: Conference Series*, vol. 656, no. 1. IOP Publishing, 2015, p. 012006.
- [6] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 1999, pp. 246–252.
- [7] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Trans. on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.
- [8] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and computing*, vol. 10, no. 3, pp. 197–208, 2000.
- [9] M. Isard and A. Blake, "Condensationconditional density propagation for visual tracking," *International journal of computer vision*, vol. 29, no. 1, pp. 5–28, 1998.
- [10] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "An adaptive color-based particle filter," *Image and vision computing*, vol. 21, no. 1, pp. 99–110, 2003.

- [11] C. Yang, R. Duraiswami, and L. Davis, "Fast multiple object tracking via a hierarchical particle filter," in *Proc. of 10th IEEE Intl. Conf. on Computer Vision (ICCV)*, vol. 1, 2005, pp. 212–219.
- [12] L. Vacchetti, V. Lepetit, and P. Fua, "Combining edge and texture information for real-time accurate 3D camera tracking," in *Proc. of 3rd IEEE and ACM International Symp. on Mixed and Augmented Reality (ISMAR)*. IEEE, 2004, pp. 48–56.
- [13] V. Takala and M. Pietikainen, "Multi-object tracking using color, texture and motion," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2007, pp. 1–7.
- [14] I. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, no. 6, pp. 810–815, 2004.
- [15] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 1269–1276.
- [16] —, "Tracking by sampling trackers," in *IEEE Int'l Conf. on Computer Vision (ICCV)*. IEEE, 2011, pp. 1195–1202.
- [17] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.
- [18] J. Ye, Z. Liu, and J. Zhang, "A face tracking algorithm based on LBP histograms and particle filtering," in *Proc. of 6th International Conf. on Natural Computation (ICNC)*, vol. 7. IEEE, 2010, pp. 3550–3553.
- [19] J. Hoeting, D. Madigan, A. Raftery, and C. Volinsky, "Bayesian model averaging: A tutorial," *Statistical science*, vol. 14, no. 4, pp. 382–401, 1999.
- [20] A. Raftery, D. Madigan, and J. Hoeting, "Bayesian model averaging for linear regression models," *Journal of the American Statistical Association*, vol. 92, no. 437, pp. 179–191, 1997.
- [21] B. Wintle, M. McCarthy, C. Volinsky, and R. Kavanagh, "The use of Bayesian model averaging to better represent uncertainty in ecological models," *Conservation Biology*, vol. 17, no. 6, pp. 1579–1590, 2003.
- [22] A. Smith, A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo methods in practice*. Springer Science & Business Media, 2013.
- [23] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering methods for practitioners," *IEEE Trans. on Signal Processing*, vol. 50, no. 3, pp. 736–746, 2002.
- [24] X. Hu, T. B. Schön, and L. Ljung, "A basic convergence result for particle filtering," *IEEE Trans. on Signal Processing*, vol. 56, no. 4, pp. 1337–1348, 2008.
- [25] B. Liu, "Instantaneous frequency tracking under model uncertainty via dynamic model averaging and particle filtering," *IEEE Trans. on Wireless Communications*, vol. 10, no. 6, pp. 1810–1819, 2011.
- [26] H. Ying, X. Qiu, J. Song, and X. Ren, "Particle filtering object tracking based on texture and color," in *Proc. of Intl. Symp. on Intelligence Information Processing and Trusted Computing (IPTC)*, 2010, pp. 626–630.
- [27] L. Li, T. Nawaz, and J. Ferryman, "Pets 2015: Datasets and challenge," in *Proc. of 12th IEEE Int'l Conf. on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2015, pp. 1–6.

TABLE I: Computation time comparison in case I(unit: second)

ALG I	ALG II	ALG III	ALG IV	the proposed
3.404	5.668	7.797	9.255	8.632

TABLE II: Computation time comparison in case III (unit: second)

ALG I	ALG II	ALG III	ALG IV	the proposed
12.793	10.286	13.934	22.116	14.637

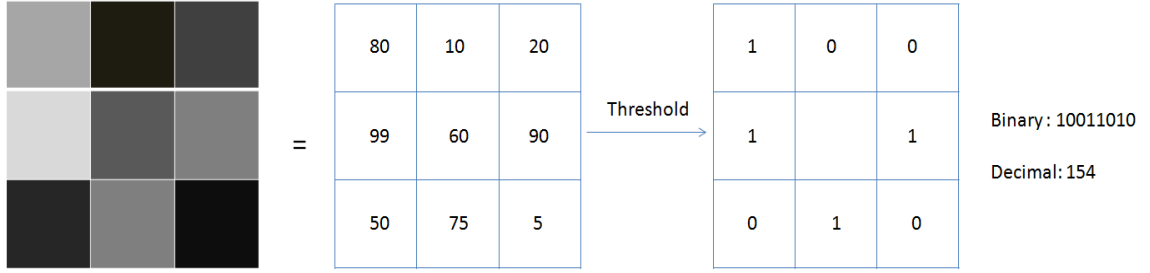


Fig. 1: A basic conceptual show of the LBP operator

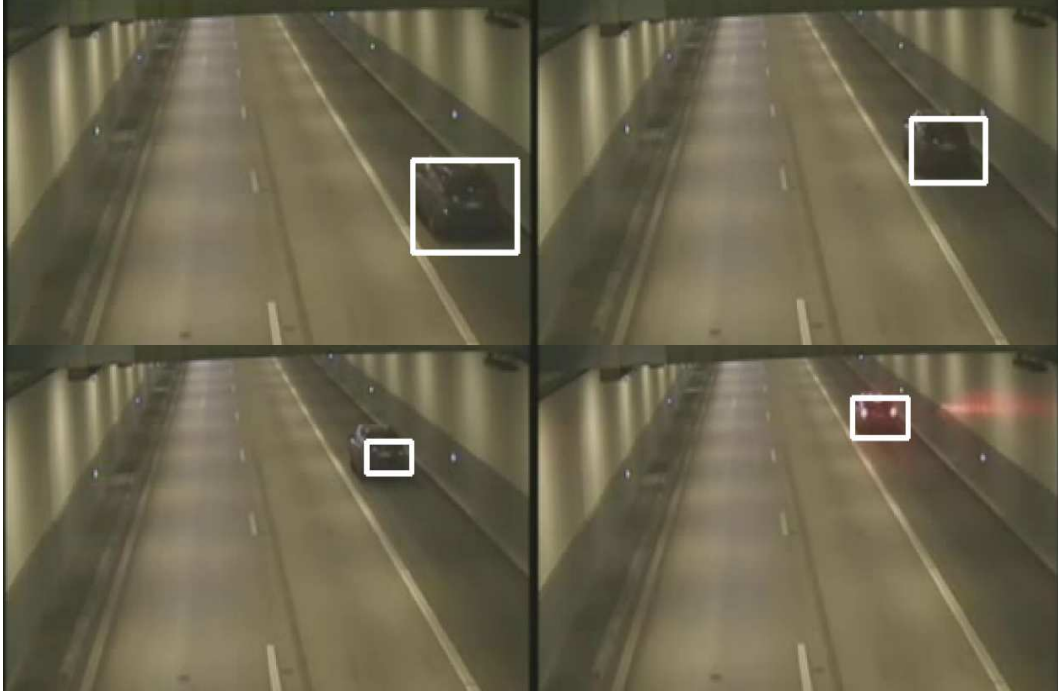


Fig. 2: Tracking results of the proposed BMA algorithm for case I. In the upper left sub-figure, this car has just entered the surveillance region. The upper right and lower left sub-figures show the middle process of the surveillance. In the lower right sub-figure, the taillights of the car has just lighted up. The white box indicates the object contour outputted by the algorithm.

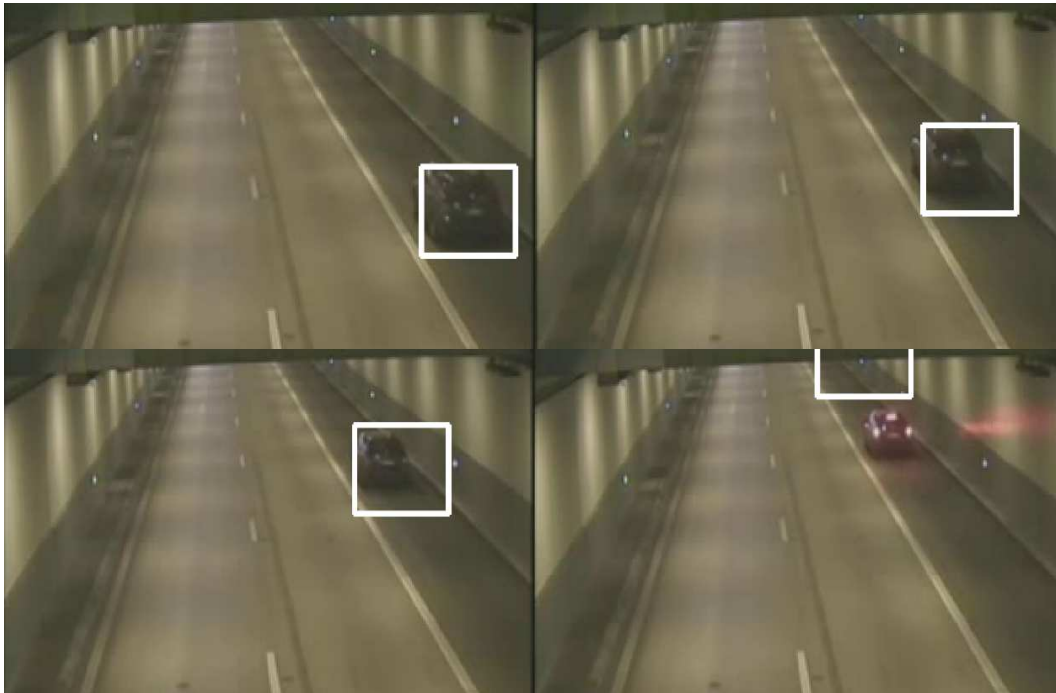


Fig. 3: Tracking results of ALG I, namely the PF tracker using adaptive color feature [10], for case I.

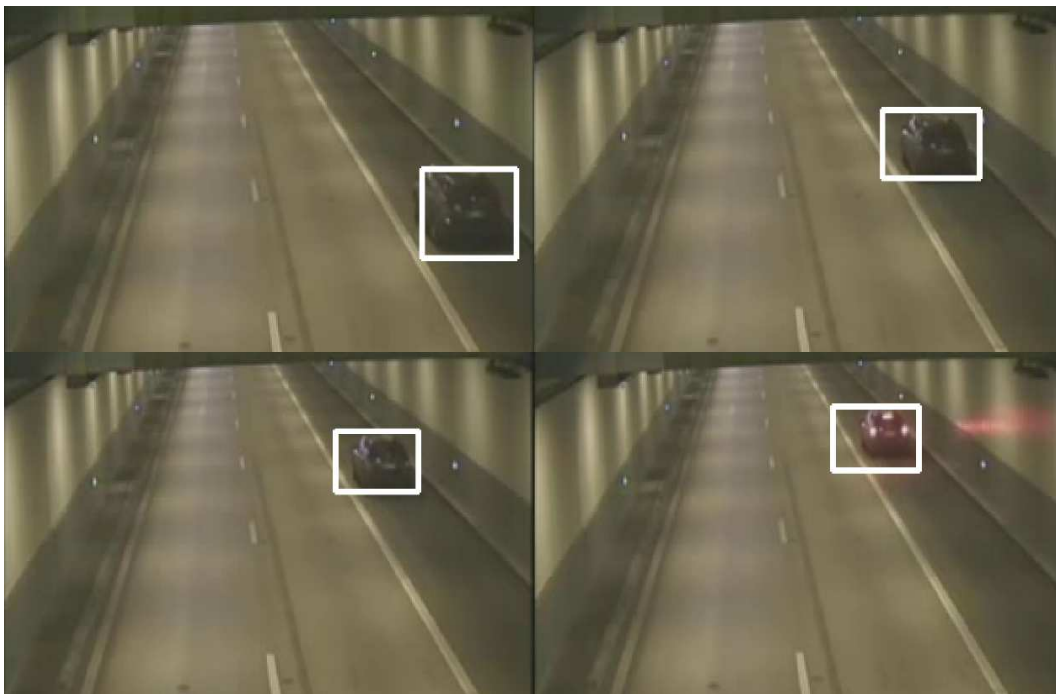


Fig. 4: Tracking results of ALG II, namely the PF tracker using LBP modeled texture feature [18], for case I.

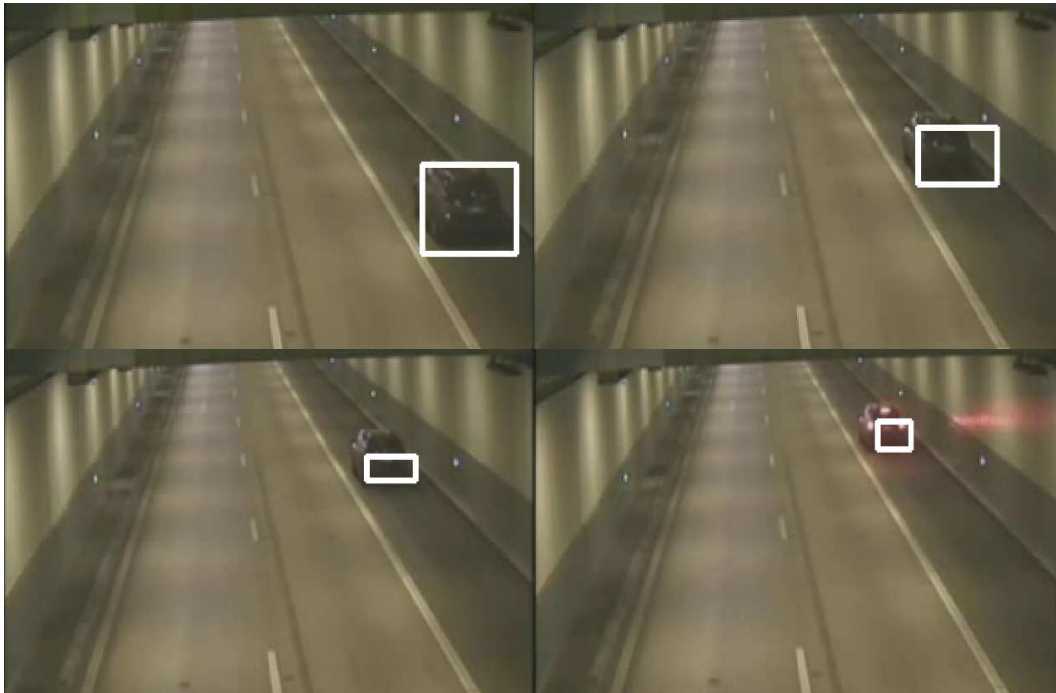


Fig. 5: Tracking results of ALG III, namely the PF tracker using equally weighted texture and color features [26], for case I.

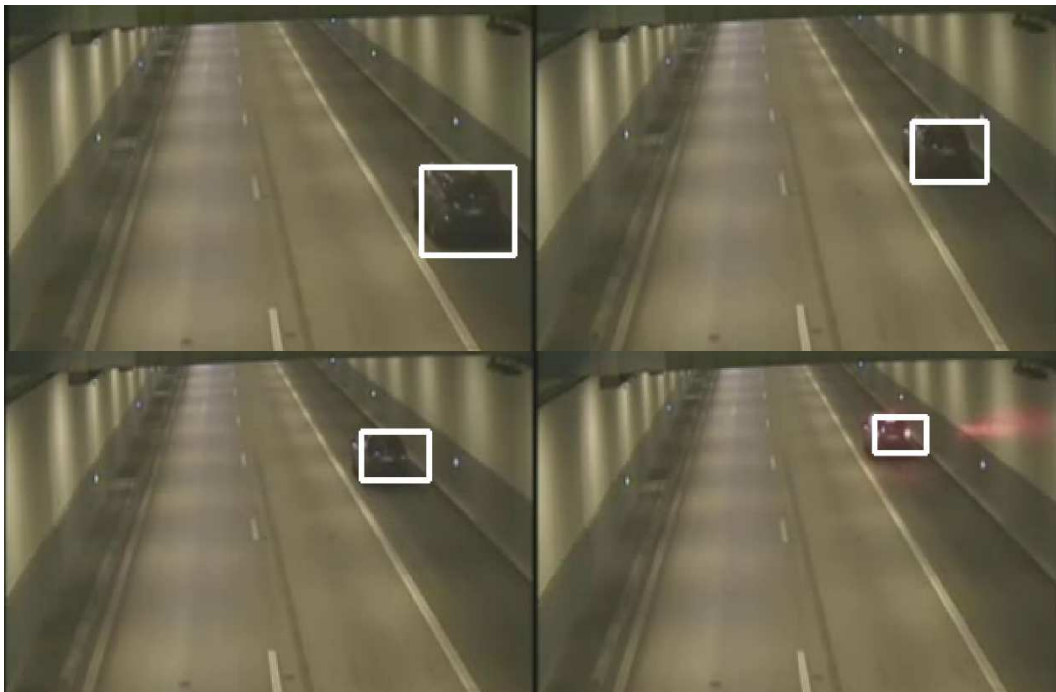


Fig. 6: Tracking results of ALG IV, namely the adaptive GM detector [6], for case I.

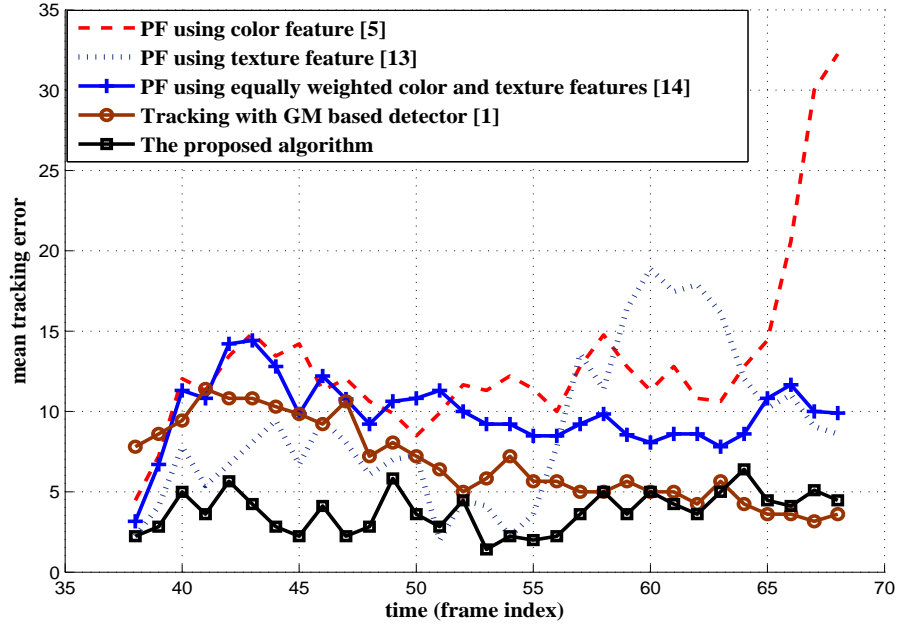


Fig. 7: Mean tracking error in test case I. The object appears in the 38th frame. The size of a whole image in one frame is 320×240 .

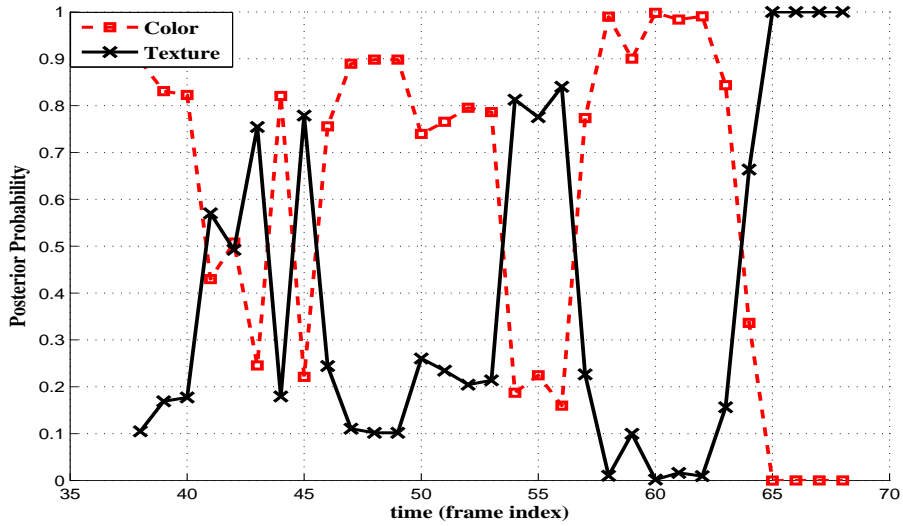


Fig. 8: Posterior probabilities of the involved feature models in test case I.

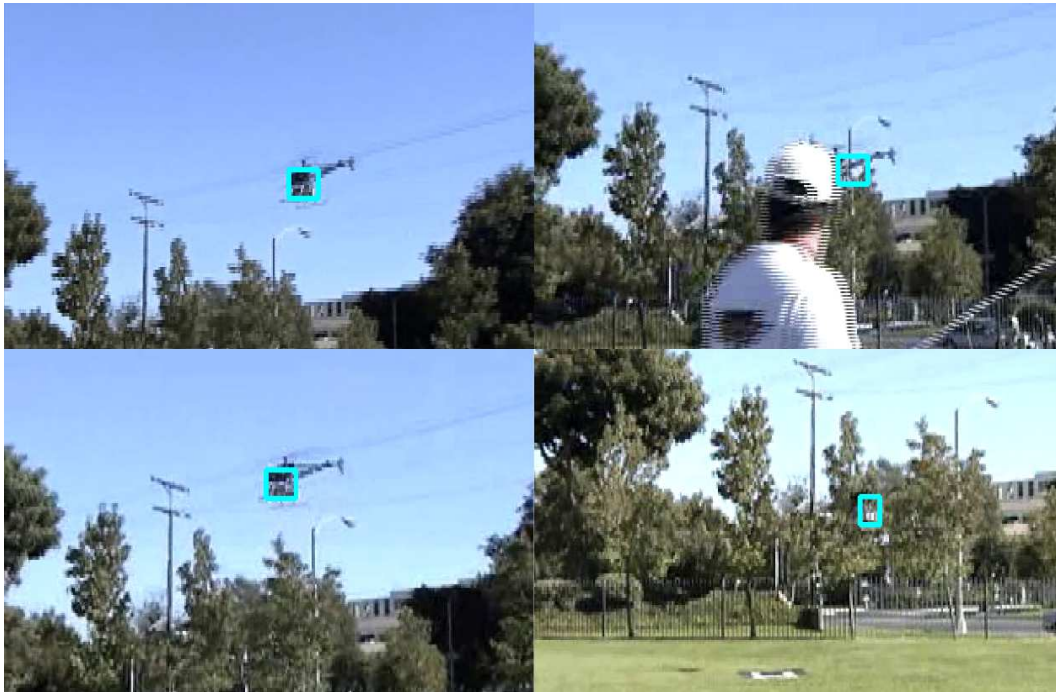


Fig. 9: Tracking results of the proposed BMA algorithm for case II

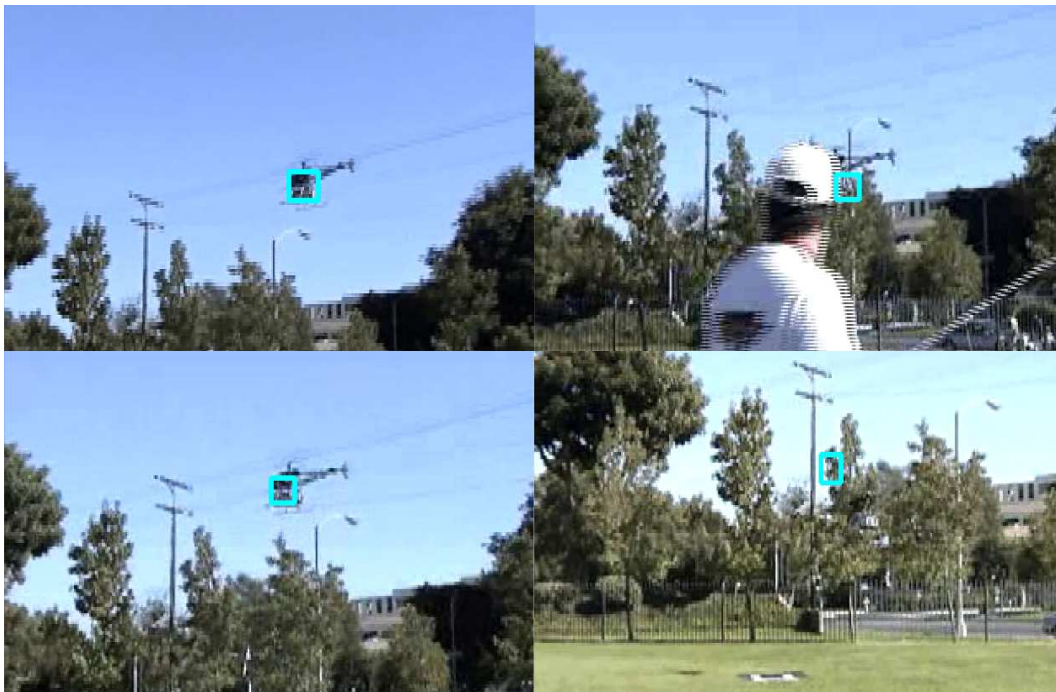


Fig. 10: Tracking results of ALG I, namely PF tracker using adaptive color feature [10], for case II

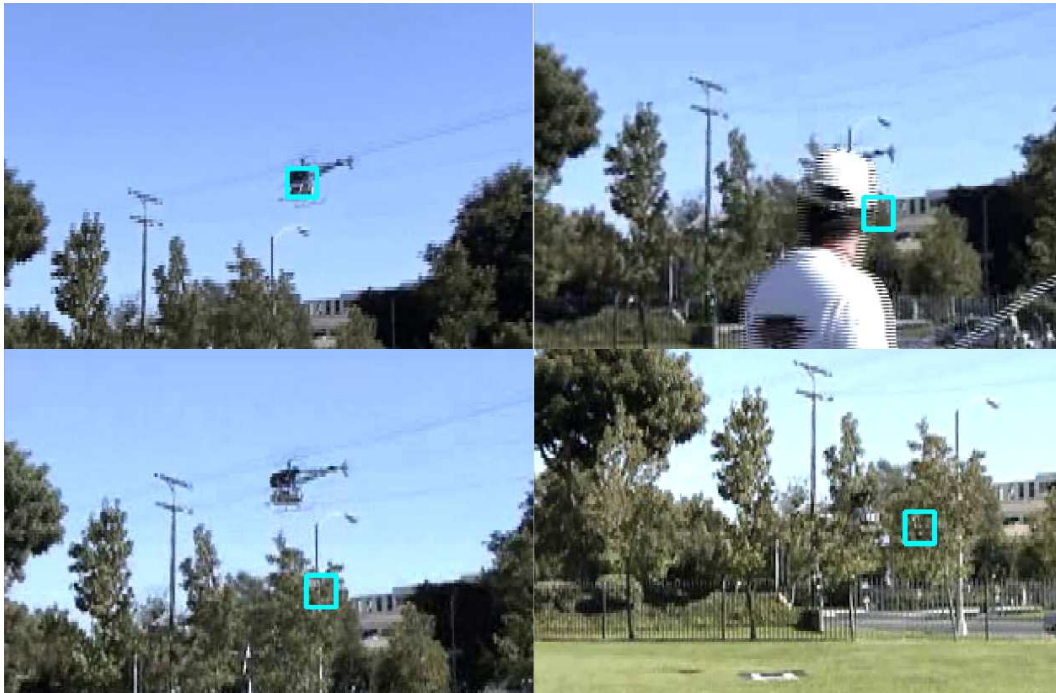


Fig. 11: Tracking results of ALG II, namely the PF tracker using the LBP modeled texture feature [18], for case II

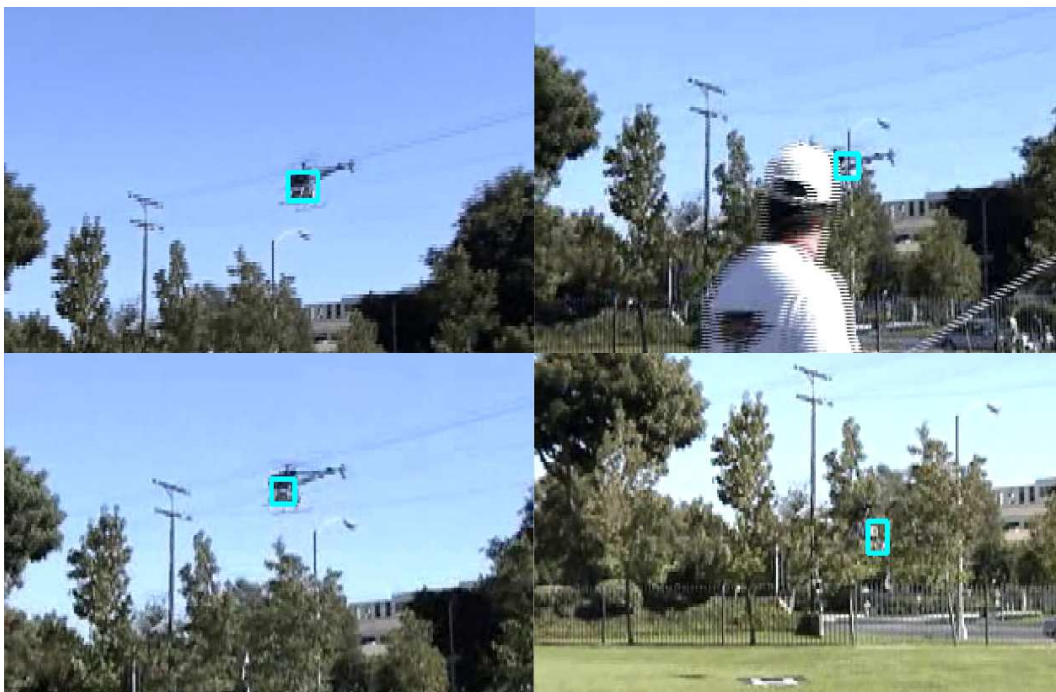


Fig. 12: Tracking results of ALG III, namely the PF tracker using equally weighted texture and color features [26], for case II

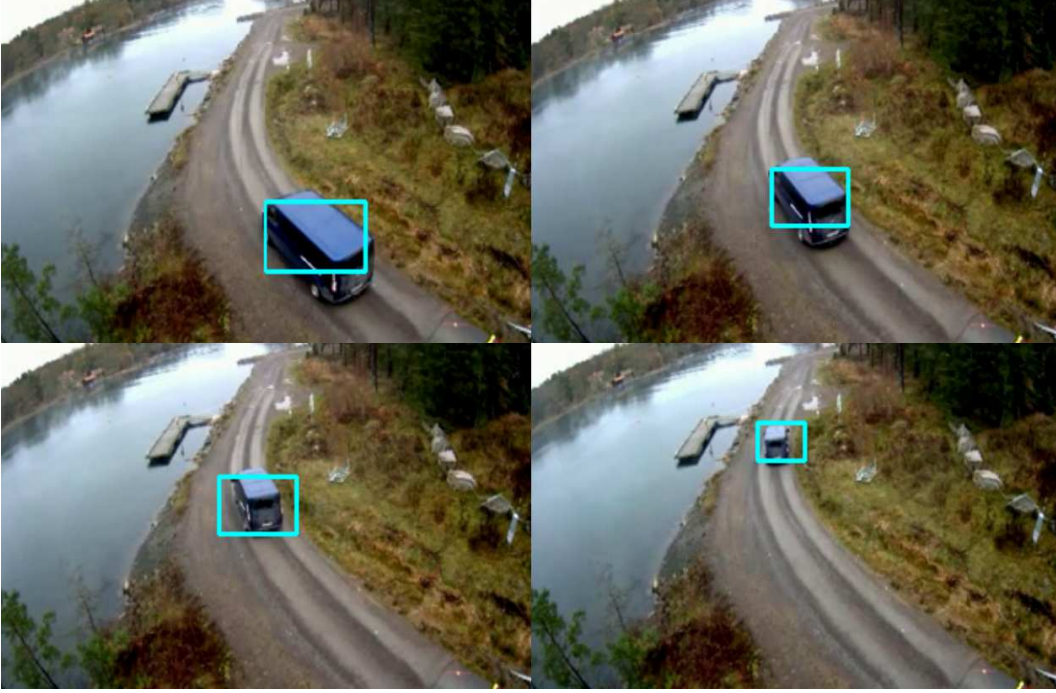


Fig. 13: Tracking results of the proposed algorithm for case III. The top left, top right, bottom left and bottom right sub-figures correspond to the 300th, 340th, 420th and 460th frames, respectively.

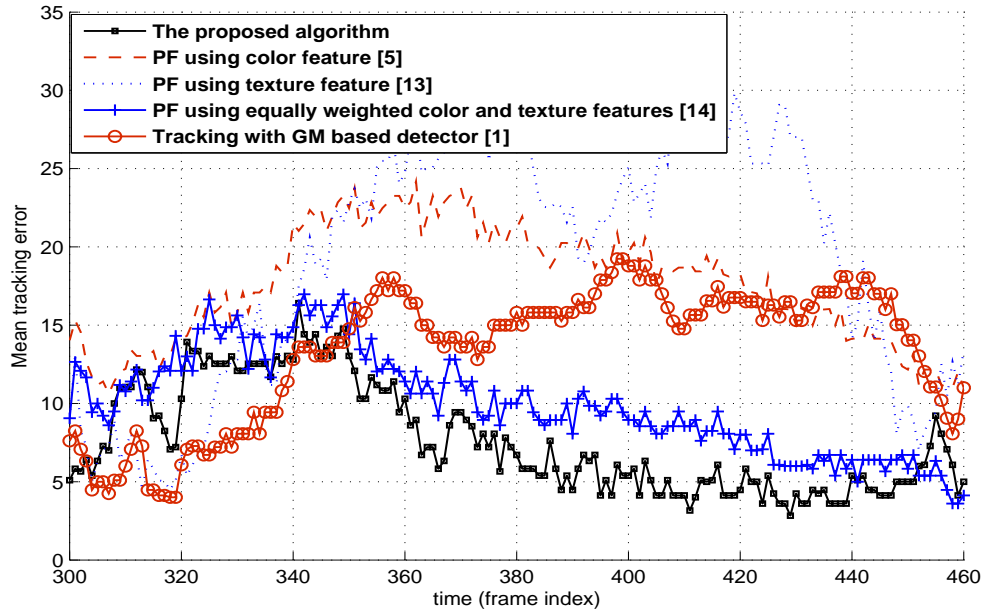


Fig. 14: Mean tracking error in test case III.

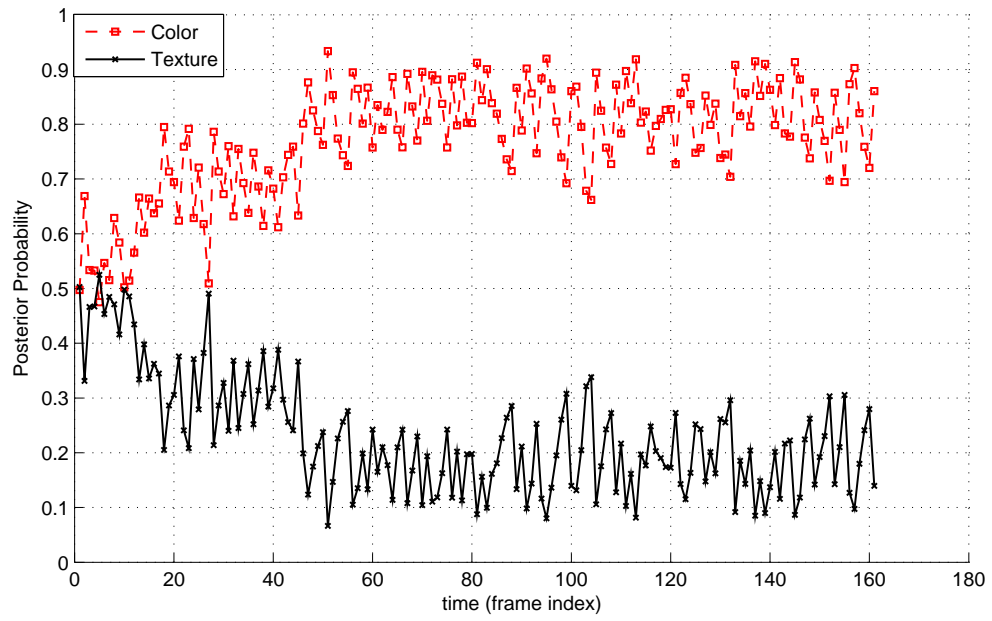


Fig. 15: Posterior probabilities of the involved feature models at each time step in test case III.